

Kolačići i sesije

Kolačići i sesije su pojmovi koji se često koriste zajedno kada je reč o internet sajtovima i aplikacijama, s tim što su kolačići uskladišteni na klijentskoj strani, a podaci o sesiji se mogu čuvati i na serverskoj i na klijentskoj strani. Ova dva pojma su najviše vezana za aplikativne protokola, a ponajviše za HTTP.

Kolačići (Cookies)

Kolačić je mali deo teksta koji je uskladišten na klijentskoj strani, najčešće od strane internet pregledača, a poslao ga je server. Kolačić kao takav, uskladišten na klijentskom uređaju, može se po potrebi slati serveru. Uobičajena upotreba kolačića se može naći pri autentifikaciji klijenta sa serverom, skladištenju korisničkih preferencija – kao što su, na primer proizvodi smešteni u korpu prilikom elektronske kupovine; pri identifikaciji klijenta od strane servera, skoro posećenih stranica itd. Takođe, ne čuvaju se ni svi kolačići koje server šalje klijentu – već po potrebi. Neki kolačići traju dokle god postoji aktivno pretraživanje konkretnog sajta ili aplikacije od strane klijenta u trenutnom vremenu. Kolačići se kreiraju u formatu ime kolačića – vrednost kolačića.

Kad god se odigrava komunikacija između klijenta i servera – pregledač će uglavnom slati serveru informacije o postojećim kolačićima na našem uređaju. Biće poslato samo kolačići koje je uskladištio taj pregledač (ili bilo koji drugi klijent) i koji se odnose na serverski domen. Ovo znači da će, ako korisnik pomoću pregledača pristupa sajtu www.example.com, isti taj pregledač prilikom tog zahteva poslati samo kolačiće koje poseduje koji su povezani sa tim sajtom. U suštini, kolačići su dobar način za povezivanje jedne stranice sa drugom prilikom korisničke interakcije sa web sajtom ili aplikacijom.

Još jedan bitan aspekt kolačića je i sigurnost. Iako kolačići po svojoj prirodi nisu štetni, nisu retki slučajevi da su kolačići bili iskorišćavani za prenos virusa i ostalih zlonamernih programa i kodova. Sa razvojem pregledača, ovakav scenario je danas ređe mogući.

Sesija

Sesija se može definisati kao serversko skladištenje informacija o klijentu koje postoji dokle god klijent ne napusti server ili serversku aplikaciju. Ideja iza sesija je da se, umesto skladištenja velikog broja podataka koji se često razmenjaju preko kolačića u klijentu, jedino jedinstveni identifikator uskladišti na klijentskoj strani (ovakav identifikator se uglavnom zove session id). Ovaj identifikator se prosleđuje serveru kad god klijent pošalje zahtev serveru. Na ovaj način, web aplikacija uparuje ovaj identifikacioni broj sesije sa svojom internom bazom kako bi pronašao podatke koji se tiču dotičnog klijenta (ako postoje).

Postavljanje kolačića pomoću requests biblioteke

Rukovanje kolačićima u Pythonu ćemo pokazati koristeći, umesto pregledača, requests biblioteku sa kojom smo se upoznali u kursu *Python Net Programming*, koju je prethodno potrebno instalirati pomoću pip alata, jer nije deo ugrađenih biblioteka, dok ćemo kao testni sajt koristiti httpbin.org. Ovaj sajt podržava postavljanje, čitanje i brisanje kolačića na sledećim linkovima:

- <https://httpbin.org/cookies> – čitanje postojećih kolačića;
- <https://httpbin.org/cookies/set> – postavljanje novih kolačića (zbog načina na koji je sajt httpbin.org definisan, kolačići se mogu proslediti i kroz sam link: https://httpbin.org/cookies/set/ime_kolačića/vrednost_kolačića);
- <https://httpbin.org/cookies/delete> – brisanje postojećih kolačića.

Ako bismo napravili GET zahtev ka <https://httpbin.org/cookies>, dobili bismo prazan odgovor od servera, jer trenutno ne postoje kolačići koje bi nam server mogao vratiti. Zato ćemo iskoristiti <https://httpbin.org/cookies/set> stranu, kojoj ćemo poslati set kolačića. Sami kolačići koji se šalju preko requests biblioteke su tipa rečnik i takvi se prosleđuju argumentu cookies poziva get().

Primer slanja kolačića uz zahtev na server

```
import requests
r = requests.get('https://httpbin.org/cookies/set',
cookies={'cookie':'is_working'})
print(r.text)
```

Ispis:

```
{  "cookies": {"cookie": "is_working"  }}
```

Pošto je httpbin.org sajt podešen tako da mu povratna vrednost bude zapravo vrednost zahteva koji pošaljemo, tako nam je i vratio vrednost koja zapravo predstavlja kolačiće koje smo mi kao klijent i poslali. U realnom scenariju, kada bismo slali zahtev ka željenom sajtu, kolačićima koje bi nam taj server poslao bismo pristupili koristeći metodu cookies, koja vraća RequestsCookieJar[] objekat, dakle r.cookies. Ovaj objekat je baziran na rečnicima i predstavlja listu ključ:vrednost parametara, gde je ključ – ime kolačića, a vrednost – vrednost tog kolačića. Pa tako, ako znamo unapred ime kolačića koji server vraća, možemo mu pristupiti i po imenu: "r.cookies['cookie']".

Nakon našeg poziva requests.get('https://httpbin.org/cookies/set', cookies={'cookie':'is_working'}), poslali smo kolačiće na server. Ako bismo iskorisili <https://httpbin.org/cookies> stranicu za čitanje postojećih stranica naredbom:

```
r = requests.get("https://httpbin.org/cookies")
print(r.text)
```

opet bismo dobili prazan odgovor opet. Ovo se dešava jer sada ovim pozivom kreiramo novu sesiju (prethodna sesija je počela pozivom .get() metode nad <https://httpbin.org/cookies/set> linkom i završila se nakon tog izvršavanja). Kako nismo sačuvali nijedan trag o prethodnoj sesiji, ona prestaje da postoji i svakim narednim zahtevom kreiramo novu sesiju. Ovakav princip kreiranja novih sesija odmah nakon izvršenja zahteva je analogan korišćenju pregledača kada korisnik otvori određeni sajt, pretražuje ga i nakon toga zatvori pregledač. Da bismo u našem primeru sačuvali informaciju o poslatim kolačićima – potrebno je koristiti sesiju.

CookieJar

Biblioteka requests nam pruža još jedan pogodan način za rad sa kolačićima: Cookie Jar – bazu kolačića. Ovakvu bazu kreiramo pomoću naredbe requests.cookies.RequestsCookieJar(), nakon koje dobijamo objekat tipa <RequestsCookieJar[]> sa kojim možemo detaljnije upravljati kolačićima. Za postavljanje kolačića u ovakvu jednu predefinisanu bazu kolačića koristimo metodu .set() ovog objekta, kojoj prosleđujemo četiri parametra, gde prva dva predstavljaju ime i vrednost kolačića, dok treći parametar predstavlja domen na koji će se naš kolačić poslati, a četvrti parametar je krajnja putanja za dati domen. Kreiranje Cookie Jar objekta, konfiguracija i slanje ka serveru izgledaju ovako:

Primer korišćenja CookieJar objekta

```
import requests
jar = requests.cookies.RequestsCookieJar()
jar.set('cookie', 'one', domain='httpbin.org', path='/cookies')
jar.set('cookie', 'two', domain='httpbin.org', path='/unknown')

r = requests.get('https://httpbin.org/cookies', cookies=jar)
print(r.text)
```

Ispis:

```
{ "cookies": { "cookie": "one" } }
```

Na ovaj način olakšavamo praćenje kolačića i rukovanje kolačićima kada moramo slati više različitih zahteva za različite domene ili stranice. U našem primeru smo u bazu kolačića postavili dva kolačića: jedan koji će se proslediti kada se naš klijent poveže na httpbin.org/cookies i drugi koji će se proslediti kada se naš klijent poveže na httpbin.org/unknown. Ova druga varijanta sa putanjom /unknown ne postoji na serveru i korišćena je kao primer.

Pitanje

Kolačići se ključnom argumentu cookies metode .get() prosleđuju kao tip:

- rečnik
- string
- n-torka

Objašnjenje:

Tačan odgovor je da se kolačići ključnom argumentu cookies metode .get() prosleđuju kao tip rečnik, gde je ključ ime kolačića, a vrednost ključa – vrednost kolačića koji šaljemo.

Sesije pomoću requests biblioteke

Biblioteka requests nam pruža jednostavan način upravljanja sesijama i kreiranja sesijskog objekta. Sesije u requests biblioteci su implementirane tako da se ista TCP konekcija koristi za slanje svih zahteva ka tom jednom domenu. Pri promeni domena, kreira se nova TCP konekcija.

Korišćenjem sesija održavamo trajnost određenih parametara, koja se, osim na kolačiće, odnosi i na druge parametre u zahtevu (primer: zaglavlje). Za kreiranje sesijskog objekta koristimo metodu `Session()` ove biblioteke. Metode koje smo pre koristili važe i nad sesijskim objektom (`get`, `post`, `cookies`...).

U prethodnom primeru smo jednim `get()` pozivom nad linkom <https://httpbin.org/cookies/set> poslali željene kolačiće, a drugim pozivom `get()` metode nad linkom <https://httpbin.org/cookies> pokušali da ih iščitamo. Kako nije bilo reči o sesijskoj konekciji, prilikom iščitavanja dobili smo prazan odgovor. Prethodni primer u slučaju korišćenja sesije izgleda ovako:

Primer slanja GET zahteva sa argumentima

```
import requests
first_session = requests.Session()
first_session.cookies.update({'default_cookie': 'is_default'})
req_one =
first_session.get('https://httpbin.org/cookies/set/cookie/is_working')
req_two = first_session.get('http://httpbin.org/cookies')
print("Second request response: ", req_two.text)
```

Ispis:

```
Second request response: {
  "cookies": {
    "cookie": "is_working",
    "default_cookie": "is_default"  }}
```

Ovde lako uviđamo razliku u odnosu na prethodnu varijantu. Naime, kako bismo održali trajnost parametara kroz pozive u okviru sesije, koristimo naredbu `first_session.cookies.update()`, kojoj prosleđujemo tip rečnik sa ključ:vrednost parovima kolačića koje želimo da prosledimo prilikom svakog poziva. Dalje, u prvom pozivu postavljamo kolačić korišćenjem httpbin.org/cookies/set linka. Nakon ovog poziva, naš sesijski objekat nam omogućava da održavamo istu ovu konekciju sa serverom. Pa tako, nakon `get` poziva nad linkom <https://httpbin.org/cookies> i analize odgovora servera uviđamo da su se kolačići zaista održali i sačuvali i nakon drugog zahteva. Na ovaj način se i zaglavlja mogu slati i pamtit u okviru sesije.

Rezime

- Kolačić je mali deo teksta koji je uskladišten na klijentskoj strani od strane, najčešće, internet pregledača, a poslao ga je strane servera.
- Uobičajena upotreba kolačića se može naći pri autentifikaciji klijenta sa serverom, skladištenju korisničkih preferencija kao što su, na primer, proizvodi smešteni u korpu prilikom elektronske kupovine, identifikaciji klijenta od strane servera, skoro posećenih stranica itd.
- Kolačići su dobar način za povezivanje jedne stranice sa drugom prilikom korisničke interakcije sa web sajtom ili aplikacijom.
- Sesija se može definisati kao serversko skladištenje informacija o klijentu koje postoji dokle god klijent ne napusti server ili serversku aplikaciju.
- Za slanje kolačića korišćenjem requests biblioteke, pri zahtevu (get, post, put...) koristimo ključni argument cookies, kome prosleđujemo rečnik sa ključ:vrednost parovima željenih kolačića.
- CookieJar je funkcionalnost requests biblioteke preko koje možemo detaljnije upravljati kolačićima i raspoređivati ih prema domenima i stranicama na koje želimo da ih pošaljemo.
- Sesije se pomoću requests biblioteke kreiraju preko klasnog konstruktora Session() biblioteke requests.
- Sesije u requests biblioteci su implementirane tako da se ista TCP konekcija koristi za slanje svih zahteva ka tom jednom domenu.

